Apache Spark - Practice set 2

Total points44/60

The respondent's email (baburam@fusemachines.com) was recorded on submission of this form.

Q 1. Which of the following statements about the Spark driver is incorrect?

1/1

A. The Spark driver is the node in which the Spark application's main method runs to coordinate the Spark application.

B. The Spark driver is horizontally scaled to increase overall processing throughput

Correct

D. The Spark driver is responsible for scheduling the execution of data by various worker nodes in cluster mode.

E. The Spark driver should be as close as possible to worker nodes for optimal performance.

Q 2. Which of the following describes nodes in cluster-mode Spark?

1/1

A. Nodes are the most granular level of execution in the Spark execution hierarchy.

B. There is only one node and it hosts both the driver and executors.

C. Nodes are another term for executors, so they are processing engine instances for performing computations.

D. There are driver nodes and worker nodes, both of which can scale horizontally.

E. Worker nodes are machines that host the executors responsible for the execution of tasks.

Correct

Q 3. Which of the following statements about slots is true?

1/1

A. There must be more slots than executors.

B. There must be more tasks than slots.

C. Slots are the most granular level of execution in the Spark execution hierarchy.

D. Slots are not used in cluster mode.

E. Slots are resources for parallelization within a Spark application.

Correct

Q 4. Which of the following is a combination of a block of data and a set of transformers that will run on a single executor?

1/1

A. Executor

B. Node
C. Job
D. Task
Correct

E. Slot

Q 5. Which of the following is a group of tasks that can be executed in parallel to compute the same set of operations on potentially multiple machines?
1/1
A. Job
B. Slot
C. Executor
D. Task
E. Stage
Correct

Q 6. Which of the following describes a shufle?
1/1
A. A shufle is the process by which data is compared across partitions.
Correct

B. A shufle is the process by which data is compared across executors.
C. A shufle is the process by which partitions are allocated to tasks.
D. A shufle is the process by which partitions are ordered for write.
E. A shufle is the process by which tasks are ordered for execution.

Q 7. DataFrame df is very large with a large number of partitions, more than there are executors in the cluster. Based on this situation, which of the following is incorrect? Assume there is one core per executor.
1/1
A. Performance will be suboptimal because not all executors will be utilized at the same time.
Correct

B. Performance will be suboptimal because not all data can be processed at the same time.
C. There will be a large number of shufle connections performed on DataFrame df when operations inducing a shufle are called.

D. There will be a lot of overhead associated with managing resources for data processing within each task.
E. There might be risk of out-of-memory errors depending on the size of the executors in the cluster.

Q 8. Which of the following operations will trigger evaluation?
1/1
A. DataFrame.filter()
B. DataFrame.distinct()
C. DataFrame.intersect()
D. DataFrame.join()
E. DataFrame.count()
Correct

Q 9. Which of the following describes the difference between transformations and actions?
1/1
A. Transformations work on DataFrames/Datasets while actions are reserved for native language objects.
B. There is no difference between actions and transformations.
C. Actions are business logic operations that do not induce execution while transformations are execution triggers focused on returning results.
D. Actions work on DataFrames/Datasets while transformations are reserved for native language objects.
E. Transformations are business logic operations that do not induce execution while actions are execution triggers focused on returning results.
Correct

Q 10. Which of the following DataFrame operations is always classified as a narrow transformation?
1/1
A. DataFrame.sort()
B. DataFrame.distinct()
C. DataFrame.repartition()
D. DataFrame.select()
Correct

E. DataFrame.join()

Q 11. Spark has a few different execution/deployment modes: cluster, client, and local. Which of the following describes Spark's execution/deployment mode?
1/1

A. Spark's execution/deployment mode determines where the driver and executors are physically located when a Spark application is run
Correct

B. Spark's execution/deployment mode determines which tasks are allocated to which executors in a cluster
C. Spark's execution/deployment mode determines which node in a cluster of nodes is responsible for running the driver program
D. Spark's execution/deployment mode determines exactly how many nodes the driver will connect to when a Spark application is run
E. Spark's execution/deployment mode determines whether results are run interactively in a notebook environment or in batch

Q 12. Which of the following cluster configurations will ensure the completion of a Spark application in light of a worker node failure? Note: each configuration has roughly the same compute power using 100GB of RAM and 200 cores.
1/1
A. Scenario #1
B. They should all ensure completion because worker nodes are fault-tolerant.
Correct

C. Scenario #4
D. Scenario #5
E. Scenario #6

Q 13. Which of the following describes out-of-memory errors in Spark?
1/1
A. An out-of-memory error occurs when either the driver or an executor does not have enough memory to collect or process the data allocated to it.
Correct

B. An out-of-memory error occurs when Spark's storage level is too lenient and allows data objects to be cached to both memory and disk.
C. An out-of-memory error occurs when there are more tasks than are executors regardless of the number of worker nodes.
D. An out-of-memory error occurs when the Spark application calls too many transformations in a row without calling an action regardless of the size of the data object on which the transformations are operating.
E. An out-of-memory error occurs when too much data is allocated to the driver for computational purposes.

Q 14. Which of the following is the default storage level for persist() for a non-streaming DataFrame/Dataset?
1/1

A. MEMORY_AND_DISK
Correct

B. MEMORY_AND_DISK_SER
C. DISK_ONLY
D. MEMORY_ONLY_SER
E. MEMORY_ONLY

Q 15. Which of the following describes a broadcast variable?
1/1
A. A broadcast variable is a Spark object that needs to be partitioned onto multiple worker nodes because it's too large to fit on a single worker node.
B. A broadcast variable can only be created by an explicit call to the broadcast() operation.
C. A broadcast variable is entirely cached on the driver node so it doesn't need to be present on any worker nodes.
D. A broadcast variable is entirely cached on each worker node so it doesn't need to be shipped or shufled between nodes with each stage.
Correct

E. A broadcast variable is saved to the disk of each worker node to be easily read into memory when needed.

Q 16. Which of the following operations is most likely to induce a skew in the size of your data's partitions?
1/1
A. DataFrame.collect()
B. DataFrame.cache()
C. DataFrame.repartition(n)
D. DataFrame.coalesce(n)
Correct

E. DataFrame.persist()

Q 17. Which of the following data structures are Spark DataFrames built on top of?
1/1
A. Arrays
B. Strings
C. RDDs
Correct

D. Vectors
E. SQL Tables

Q 18. Which of the following code blocks returns a DataFrame containing only column storeId and column division from DataFrame storesDF?
1/1
A. storesDF.select("storeId").select("division")
B. storesDF.select(storeId, division)
C. storesDF.select("storeId", "division")
Correct

D. storesDF.select(col("storeId", "division"))
E. storesDF.select(storeId).select(division)

Q 19. Which of the following code blocks returns a DataFrame containing all columns from DataFramestoresDF except for column sqft and column customerSatisfaction? A sample of DataFrame storesDF is below:
1/1
A. storesDF.drop("sqft", "customerSatisfaction")
Correct

B. storesDF.select("storeId", "open", "openDate", "division")
C. storesDF.select(-col(sqft), -col(customerSatisfaction))
D. storesDF.drop(sqft, customerSatisfaction)
E. storesDF.drop(col(sqft), col(customerSatisfaction))

Q 20. The below code shown block contains an error. The code block is intended to return a DataFrame containing only the rows from DataFrame storesDF where the value in DataFrame storesDF's "sqft" column is less than or equal to 25,000. Assume DataFrame storesDF is the only defined language variable. Identify the error. Code block: storesDF.filter(sqft <= 25000)
1/1
A. The column name sqft needs to be quoted like storesDF.filter("sqft" <= 25000).
B. The column name sqft needs to be quoted and wrapped in the col() function like storesDF.filter(col("sqft") <= 25000).
Correct

C. The sign in the logical condition inside filter() needs to be changed from <= to >.
D. The sign in the logical condition inside filter() needs to be changed from <= to >=.
E. The column name sqft needs to be wrapped in the col() function like storesDF.filter(col(sqft) <= 25000).

Q 21. The code block shown below should return a DataFrame containing only the rows from DataFrame storesDF where the value in column sqft is less than or equal to 25,000 OR the value in column customerSatisfaction is greater than or equal to 30. Choose the response that correctly fills in the numbered blanks within the code block to complete this task. Code block: storesDF.__1__(__2__ __3__ __4__)

0/1

A.1.filter 2. (col("sqft") <= 25000) 3. | 4. (col("customerSatisfaction") >= 30)

B.1. drop 2. (col(sqft) <= 25000) 3. | 4. (col(customerSatisfaction) >= 30)

Incorrect

C.1. filter 2. col("sqft") <= 25000 3. | 4. col("customerSatisfaction") >= 30

D. 1. Filter 2. col("sqft") <= 25000 3.Or 4.col("customerSatisfaction") >= 30

E. 1. Filter 2. (col("sqft") <= 25000) 3.Or 4. (col("customerSatisfaction") >= 30)

Correct answer

A.1.filter 2. (col("sqft") <= 25000) 3. | 4. (col("customerSatisfaction") >= 30)

Q 22. Which of the following operations can be used to convert a DataFrame column from one type to another type?

1/1

A. col().cast()

Correct

B. convert()

C. castAs()

D. col().coerce()

E. col()

Q 23. Which of the following code blocks returns a new DataFrame with a new column sqft100 that is 1/100th of column sqft in DataFrame storesDF? Note that column sqft100 is not in the original DataFrame storesDF.

1/1

A. storesDF.withColumn("sqft100", col("sqft") * 100)

B. storesDF.withColumn("sqft100", sqft / 100)

C. storesDF.withColumn(col("sqft100"), col("sqft") / 100)

D. storesDF.withColumn("sqft100", col("sqft") / 100)

Correct

E. storesDF.newColumn("sqft100", sqft / 100)

Q 24. Which of the following code blocks returns a new DataFrame from DataFrame storesDF where column numberOfManagers is the constant integer 1?
1/1
A. storesDF.withColumn("numberOfManagers", col(1))
B. storesDF.withColumn("numberOfManagers", 1)
C. storesDF.withColumn("numberOfManagers", lit(1))
Correct

D. storesDF.withColumn("numberOfManagers", lit("1"))
E. storesDF.withColumn("numberOfManagers", IntegerType(1))

Q 25. The code block shown below contains an error. The code block intends to return a new DataFrame where column storeCategory from DataFrame storesDF is split at the underscore character into column storeValueCategory and column storeSizeCategory. Identify the error.A sample of DataFrame storesDF is displayed below:Code block:(storesDF.withColumn("storeValueCategory", col("storeCategory").split("_")[0]).withColumn("storeSizeCategory", col("storeCategory").split("_")[1]))
1/1
A. The split() operation comes from the imported functions object. It accepts a string column name and split character as arguments. It is not a method of a Column object.
B. The split() operation comes from the imported functions object. It accepts a Column object and split character as arguments. It is not a method of a Column object.
Correct

C. The index values of 0 and 1 should be provided as second arguments to the split() operation rather than indexing the result.
D. The index values of 0 and 1 are not correct — they should be 1 and 2, respectively.
E. The withColumn() operation cannot be called twice in a row.

Q 26. Which of the following operations can be used to split an array column into an individual DataFrame row for each element in the array?
1/1
A. extract()
B. split()
C. explode()
Correct

D. arrays_zip()
E. unpack()

Q 27. Which of the following code blocks returns a new DataFrame where column storeCategory is an all-lowercase version of column storeCategory in DataFrame storesDF? Assume DataFrame storesDF is the only defined language variable.

0/1

A. storesDF.withColumn("storeCategory", lower(col("storeCategory")))

B. storesDF.withColumn("storeCategory", col("storeCategory").lower())

C. storesDF.withColumn("storeCategory", tolower(col("storeCategory")))

D. storesDF.withColumn("storeCategory", lower("storeCategory"))

E. storesDF.withColumn("storeCategory", lower(storeCategory))

Incorrect

Correct answer

A. storesDF.withColumn("storeCategory", lower(col("storeCategory")))

Q 28. The code block shown below contains an error. The code block is intended to return a new DataFrame where column division from DataFrame storesDF has been renamed to column state and column managerName from DataFrame storesDF has been renamed to column managerFullName. Identify the error.Code block: (storesDF.withColumnRenamed("state", "division").withColumnRenamed("managerFullName", "managerName"))

0/1

A. Both arguments to operation withColumnRenamed() should be wrapped in the col() operation.

B. The operations withColumnRenamed() should not be called twice, and the first argument should be ["state", "division"] and the second argument should be ["managerFullName", "managerName"].

C. The old columns need to be explicitly dropped.

Incorrect

D. The first argument to operation withColumnRenamed() should be the old column name and the second argument should be the new column name.

E. The operation withColumnRenamed() should be replaced with withColumn().

Correct answer

D. The first argument to operation withColumnRenamed() should be the old column name and the second argument should be the new column name.

Q 29. Which of the following code blocks returns a DataFrame where rows in DataFrame storesDFcontaining missing values in every column have been dropped?

0/1

A. storesDF.nadrop("all")

B. storesDF.na.drop("all", subset = "sqft")

C. storesDF.dropna()

Incorrect

D. storesDF.na.drop()

E. storesDF.na.drop("all")

Correct answer

E. storesDF.na.drop("all")

Q 30. Which of the following operations fails to return a DataFrame where every row is unique?

0/1

A. DataFrame.distinct()

B. DataFrame.drop_duplicates(subset = None)

C. DataFrame.drop_duplicates()

D. DataFrame.dropDuplicates()

Incorrect

E. DataFrame.drop_duplicates(subset = "all")

Correct answer

E. DataFrame.drop_duplicates(subset = "all")

Q 31. Which of the following code blocks will not always return the exact number of distinct values in column division?

0/1

A. storesDF.agg(approx_count_distinct(col("division")).alias("divi sionDistinct"))

B. storesDF.agg(approx_count_distinct(col("division"), 0).alias("divisionDistinct"))

C. storesDF.agg(countDistinct(col("division")).alias("divisionDist inct"))

D. storesDF.select("division").dropDuplicates().count()

E. storesDF.select("division").distinct().count()

Q 32. The code block shown below should return a new DataFrame with the mean of column sqft from DataFrame storesDF in column sqftMean. Choose the response that correctly fills in the numbered blanks within the code block to complete this task.Code block:storesDF.__1__(__2__(__3__).alias("sqftMean"))

0/1

A. 1.Agg 2.Mean 3.col("sqft")

B.1.Mean 2.Col 3."sqft"

Incorrect

C.1.withColumn 2.Mean 3.col("sqft")

D.1.Agg 2.Mean 3."sqft"

E.1.Agg 2.Average 3.col("sqft")

Correct answer

A. 1.Agg 2.Mean 3.col("sqft")

Q 33. Which of the following code blocks returns the number of rows in DataFrame storesDF?
0/1

A. storesDF.withColumn("numberOfRows", count())

B. storesDF.withColumn(count().alias("numberOfRows"))

C. storesDF.countDistinct()

D. storesDF.count()

E. storesDF.agg(count())

Q 34. Which of the following code blocks returns the sum of the values in column sqft in DataFramestoresDF grouped by distinct value in column division?
0/1

A. storesDF.groupBy.agg(sum(col("sqft")))

Incorrect

B. storesDF.groupBy("division").agg(sum())

C. storesDF.agg(groupBy("division").sum(col("sqft")))

D. storesDF.groupby.agg(sum(col("sqft")))

E. storesDF.groupBy("division").agg(sum(col("sqft")))

Correct answer

E. storesDF.groupBy("division").agg(sum(col("sqft")))

Q 35. Which of the following code blocks returns a DataFrame containing summary statistics only for column sqft in DataFrame storesDF?
0/1

A. storesDF.summary("mean")

B. storesDF.describe("sqft")

C. storesDF.summary(col("sqft"))

D. storesDF.describeColumn("sqft")

E. storesDF.summary()

Q 36. Which of the following operations can be used to sort the rows of a DataFrame?
1/1

A. sort() and orderBy()

Correct

B. orderby()

C. sort() and orderby()

D. orderBy()

E. sort()

Q 37. The code block shown below contains an error. The code block is intended to return a 15 percent sample of rows from DataFrame storesDF without replacement. Identify the error. Code block: storesDF.sample(True, fraction = 0.15)
1/1
A. There is no argument specified to the seed parameter.
B. There is no argument specified to the withReplacement parameter.
C. The sample() operation does not sample without replacement — sampleby() should be used instead.
D. The sample() operation is not reproducible.
E. The first argument True sets the sampling to be with replacement.
Correct

Q 38. Which of the following operations can be used to return the top n rows from a DataFrame?
1/1
A. DataFrame.n()
B. DataFrame.take(n)
Correct

C. DataFrame.head
D. DataFrame.show(n)
E. DataFrame.collect(n)

Q 39. The code block shown below should extract the value for column sqft from the first row of DataFrame storesDF. Choose the response that correctly fills in the numbered blanks within the code block to complete this task.Code block:__1__.__2__.__3__
0/1
A.1.storesDF 2.First 3.col("sqft")
B.1.storesDF 2.first 3.sqft
C.1.storesDF 2.first 3.["sqft"]
Incorrect

D.1.storesDF 2.first() 3.sqft
E. 1.storesDF 2.first() 3.col("sqft")
Correct answer
D.1.storesDF 2.first() 3.sqft

Q 40. Which of the following lines of code prints the schema of a DataFrame?
1/1
A. print(storesDF)
B. storesDF.schema

C. print(storesDF.schema())
D. DataFrame.printSchema()
Correct

E. DataFrame.schema()

Q 41. In what order should the below lines of code be run in order to create and register a SQL UDF named "ASSESS_PERFORMANCE" using the Python function assessPerformance and apply it to column customerSatistfaction in table stores? Lines of code:1. spark.udf.register("ASSESS_PERFORMANCE", assessPerformance) 2.spark.sql("SELECT customerSatisfaction, assessPerformance(customerSatisfaction) AS result FROM stores") 3. spark.udf.register(assessPerformance, "ASSESS_PERFORMANCE") 4.spark.sql("SELECT customerSatisfaction, ASSESS_PERFORMANCE(customerSatisfaction) AS result FROM stores")
0/1
A. 3, 4
B. 1, 4
C. 3, 2
D. 2
Incorrect

E. 1, 2
Correct answer
B. 1, 4

Q 42. In what order should the below lines of code be run in order to create a Python UDF assessPerformanceUDF() using the integer-returning Python function assessPerformance and apply it to column customerSatisfaction in DataFrame storesDF?Lines of code: 1. assessPerformanceUDF = udf(assessPerformance, IntegerType) 2. assessPerformanceUDF = spark.register.udf("ASSESS_PERFORMANCE", assessPerformance) 3. assessPerformanceUDF = udf(assessPerformance, IntegerType()) 4. storesDF.withColumn("result", assessPerformanceUDF(col("customerSatisfaction"))) 5. storesDF.withColumn("result", assessPerformance(col("customerSatisfaction"))) 6. storesDF.withColumn("result", ASSESS_PERFORMANCE(col("customerSatisfaction")))
1/1
A. 3, 4
Correct

B. 2, 6
C. 3, 5
D. 1, 4
E. 2, 5

Q 43. Which of the following operations can execute a SQL query on a table?
1/1
A. spark.query()
B. DataFrame.sql()
C. spark.sql()
Correct

D. DataFrame.createOrReplaceTempView()
E. DataFrame.createTempView()

Q 44. Which of the following code blocks creates a single-column DataFrame from Python list years which is made up of integers?
0/1
A. spark.createDataFrame([years], IntegerType())
Incorrect

B. spark.createDataFrame(years, IntegerType())
C. spark.DataFrame(years, IntegerType())
D. spark.createDataFrame(years)
E. spark.createDataFrame(years, IntegerType)
Correct answer
B. spark.createDataFrame(years, IntegerType())

Q 45. Which of the following operations can be used to cache a DataFrame only in Spark's memory assuming the default arguments can be updated?
1/1
A. DataFrame.clearCache()
B. DataFrame.storageLevel
C. StorageLevel
D. DataFrame.persist()
Correct

E. DataFrame.cache()

Q 46. The code block shown below contains an error. The code block is intended to return a new4- partition DataFrame from the 8-partition DataFrame storesDF without inducing a shufle. Identify the error.Code block:storesDF.repartition(4)
1/1

A. The repartition operation will only work if the DataFrame has been cached to memory.

B. The repartition operation requires a column on which to partition rather than a number of partitions.

C. The number of resulting partitions, 4, is not achievable for an 8-partition DataFrame.

D. The repartition operation induced a full shufle. The coalesce operation should be used instead.

Correct

E. The repartition operation cannot guarantee the number of result partitions.

Q 47. Which of the following code blocks will always return a new 12-partition DataFrame from the 8-partition DataFrame storesDF?

1/1

A. storesDF.coalesce(12)

B. storesDF.repartition()

C. storesDF.repartition(12)

Correct

D. storesDF.coalesce()

E. storesDF.coalesce(12, "storeId")

Q 48. Which of the following Spark config properties represents the number of partitions used in wide transformations like join()?

1/1

A. spark.sql.shuffle.partitions

Correct

B. spark.shuffle.partitions

C. spark.shuffle.io.maxRetries

D. spark.shuffle.file.buffer

E. spark.default.parallelism

Q 49. In what order should the below lines of code be run in order to return a DataFrame containing a column openDateString, a string representation of Java's SimpleDateFormat?Note that column openDate is of type integer and represents a date in the UNIX epoch format — the number of seconds since midnight on January 1st, 1970.An example of Java's SimpleDateFormat is "Sunday, Dec 4, 2008 1:05 PM". A sample of storesDF is displayed below:Lines of code:1. storesDF.withColumn("openDateString", from_unixtime(col("openDate"), simpleDateFormat))2. simpleDateFormat = "EEEE, MMM d, yyyy h:mm a"3. storesDF.withColumn("openDateString", from_unixtime(col("openDate"), SimpleDateFormat()))4. storesDF.withColumn("openDateString", date_format(col("openDate"),

simpleDateFormat)) 5. storesDF.withColumn("openDateString", date_format(col("openDate"), SimpleDateFormat()))6. simpleDateFormat = "wd, MMM d, yyyy h:mm a"
1/1
A. 2, 3
B. 2, 1
Correct

C. 6, 5
D. 2, 4
E. 6, 1

Q 50. Which of the following code blocks returns a DataFrame containing a column month, an integer representation of the month from column openDate from DataFrame storesDF?Note that column openDate is of type integer and represents a date in the UNIX epoch format — the number of seconds since midnight on January 1st, 1970.
1/1
A. storesDF.withColumn("month", getMonth(col("openDate")))
B. storesDF.withColumn("openTimestamp", col("openDate").cast("Timestamp")).withColumn("month", month(col("openTimestamp")))
Correct

C. storesDF.withColumn("openDateFormat", col("openDate").cast("Date")).withColumn("month", month(col("openDateFormat")))
D. storesDF.withColumn("month", substr(col("openDate"), 4, 2))
E. storesDF.withColumn("month", month(col("openDate")))

Q 51. Which of the following operations performs an inner join on two DataFrames?
1/1
A. DataFrame.innerJoin()
B. DataFrame.join()
Correct

C. Standalone join() function
D. DataFrame.merge()
E. DataFrame.crossJoin()

Q 52. Which of the following code blocks returns a new DataFrame that is the result of an outer join between DataFrame storesDF and DataFrame employeesDF on column storeId?
0/1
A. storesDF.join(employeesDF, "storeId", "outer")

B. storesDF.join(employeesDF, "storeId")
Incorrect

C. storesDF.join(employeesDF, "outer", col("storeId"))
D. storesDF.join(employeesDF, "outer", storesDF.storeId == employeesDF.storeId)
E. storesDF.merge(employeesDF, "outer", col("storeId"))
Correct answer
A. storesDF.join(employeesDF, "storeId", "outer")

Q 53. The below code block contains an error. The code block is intended to return a new DataFrame that is the result of an inner join between DataFrame storesDF and DataFrame employeesDF on column storeId and column employeeId which are in both DataFrames. Identify the error.Code block:storesDF.join(employeesDF, [col("storeId"), col("employeeId")])
1/1
A. The join() operation is a standalone function rather than a method of DataFrame — the join() operation should be called where its first two arguments are storesDF and employeesDF.
B. There must be a third argument to join() because the default to the how parameter is not "inner".
C. The col("storeId") and col("employeeId") arguments should not be separate elements of a list — they should be tested to see if they're equal to one another like col("storeId") == col("employeeId").
D. There is no DataFrame.join() operation — DataFrame.merge() should be used instead.
E. The references to "storeId" and "employeeId" should not be inside the col() function — removing the col() function should result in a successful join.
Correct

Q 54. Which of the following Spark properties is used to configure the broadcasting of a DataFrame without the use of the broadcast() operation?
1/1
A. spark.sql.autoBroadcastJoinThreshold
Correct

B. spark.sql.broadcastTimeout
C. spark.broadcast.blockSize
D. spark.broadcast.compress
E. spark.executor.memoryOverhead

Q 55. The code block shown below should return a new DataFrame that is the result of a cross join between DataFrame storesDF and DataFrame employeesDF. Choose the response that correctly fills in the numbered blanks within the code block to complete this task.Code block:__1__.__2__(__3__)
1/1

A.1.storesDF 2.crossJoin 3.employeesDF, "storeId"
B.1.storesDF 2.join 3.employeesDF, "cross"
C.1.storesDF 2.crossJoin 3.employeesDF, "storeId"
D.1.storesDF 2.join 3.employeesDF, "storeId", "cross"
E.1.storesDF 2.crossJoin 3.employeesDF
Correct

Q 56. Which of the following operations performs a position-wise union on two DataFrames?
1/1
A. The standalone concat() function
B. The standalone unionAll() function
C. The standalone union() function
D. DataFrame.unionByName()
E. DataFrame.union()
Correct

Q 57. Which of the following code blocks writes DataFrame storesDF to file path filePath as parquet?
1/1
A. storesDF.write.option("parquet").path(filePath)
B. storesDF.write.path(filePath)
C. storesDF.write().parquet(filePath)
D. storesDF.write(filePath)
E. storesDF.write.parquet(filePath)
Correct

Q 58. The code block shown below contains an error. The code block is intended to write DataFrame storesDF to file path filePath as parquet and partition by values in column division. Identify the error.Code block:storesDF.write.repartition("division").parquet(filePath)
0/1
A.The argument division to operation repartition() should be wrapped in the col() function to return a Column object.
B.There is no parquet() operation for DataFrameWriter — the save() operation should be used instead.
C.There is no repartition() operation for DataFrameWriter — the partitionBy() operation should be used instead.
D.DataFrame.write is an operation — it should be followed by parentheses to return a DataFrameWriter.
Incorrect

E.The mode() operation must be called to specify that this write should not overwrite existing files.

Correct answer

C.There is no repartition() operation for DataFrameWriter — the partitionBy() operation should be used instead.

Q 59. Which of the following code blocks reads a parquet at the file path filePath into a DataFrame?

0/1

A. spark.read().parquet(filePath)

B. spark.read().path(filePath, source = "parquet")

C. spark.read.path(filePath, source = "parquet")

D. spark.read.parquet(filePath)

E. spark.read().path(filePath)

Q 60. Which of the following code blocks reads JSON at the file path filePath into a DataFrame with the specified schema schema?

1/1

A.spark.read().schema(schema).format(json).load(filePath)

B.spark.read().schema(schema).format("json").load(filePath)

C.spark.read.schema("schema").format("json").load(filePath)

D.spark.read.schema("schema").format("json").load(filePath)

E.spark.read.schema(schema).format("json").load(filePath)

Correct